Project #3: Building Networks with Tensorflow and Keras

Amir Sadovnik
COSC 525: Deep Learning (Spring 2020)

---

1 Overview

In this project you will be building an apparel classifier from images. You will be experimenting with different architectures and measuring your results on each.

2 Dataset

You will be using the fashion-mnist dataset. This is a set of 70,000 gray scale images of size 28×28 associated with a label from 10 classes. More details can be found here:
https://github.com/zalandoresearch/fashion-mnist
The dataset can be read directly using keras. Look here for more details:
https://keras.io/datasets/#fashion-mnist-database-of-fashion-articles

3 Network Details

3.1 Task 1: Fully Connected Neural Network

1. Build a feed forward neural network with the following specifications:

   - Hidden layer 1: 784 neurons with hyperbolic tangent activation function in each neuron.
   - Hidden layer 2: 512 neurons, with sigmoid activation function in each of the neuron.
   - 100 neurons, with rectified linear activation function in each of the neuron.
   - Output layer: 10 neurons representing the 10 classes, and obviously softmax activation function for each of the 10 neurons.

2. Using Min-Max scaling to scale the training dataset and using the same Min and Max values from the training set scale the test dataset ($\frac{X-X_{min}}{X_{max}-X_{min}}$).

3. Using mini-batch gradient descent to optimize the loss function: "categorical cross-entropy" on the training dataset. Please run 50 epochs with batch size of 200. Also record the loss value for each of the 50 epochs. Record the time for each epoch. Generate a epoch-loss plot, and time-loss plot.

4. Evaluate the model on the test dataset:

   - print classification accuracy.
   - Print the 10-class confusion matrix.

3.2 Task 2: Small Convolutional Neural Network

1. Build a convolutional neural network with the following specifications:

   - Convolution layer having 40 feature detectors, with kernel size 5 x 5, and rectifier as the activation function, with stride 1 and no-padding.
   - A max-pooling layer with pool size 2x2.
   - Fully connected layer with 100 neurons, and rectifier as the activation function.
   - Output layer containing 10 neurons, softmax as activation function function for each of the 10 neurons.

2. Using Min-Max scaling to scale the training dataset and using the same Min and Max values from the training set scale the test dataset ($\frac{X-X_{min}}{X_{max}-X_{min}}$).

3. Using mini-batch gradient descent to optimize the loss function: "categorical cross-entropy" on the training dataset. Please run 50 epochs with batch size of 200. Also record the loss value for each of the 50 epochs. Record the time for each epoch. Generate a epoch-loss plot, and time-loss plot.

4. Evaluate the model on the test dataset:

   - print classification accuracy.
   - Print the 10-class confusion matrix.

3.3 Task 3: Bigger Convolutional Neural Network

1. Build another convolutional neural network with the following specifications:

   - Convolution layer having 48 feature detectors, with kernel size 3 x 3, and rectifier as the activation function, with stride 1 and no-padding.
   - A max-pooling layer with pool size 2x2.
   - Convolution layer having 96 feature detectors, with kernel size 3 x 3, and rectifier as the activation function, with stride 1 and no-padding.

- A max-pooling layer with pool size 2x2.
- Fully connected layer with 100 neurons, and rectifier as the activation function.
- Output layer containing 10 neurons, softmax as activation function function for each of the 10 neurons.

2. Using Min-Max scaling to scale the training dataset and using the same Min and Max values from the training set scale the test dataset ($\frac{X-X_{min}}{X_{max}-X_{min}}$).
3. Using mini-batch gradient descent to optimize the loss function: "categorical cross-entropy" on the training dataset. Please run 50 epochs with batch size of 200. Also record the loss value for each of the 50 epochs. Record the time for each epoch. Generate a epoch-loss plot, and time-loss plot.
4. Evaluate the model on the test dataset:
   - print classification accuracy.
   - Print the 10-class confusion matrix.

3.4 Task 4: Your own Convolutional Neural Network

1. Build another convolutional neural network, where you choose all the parameters to see if you can get a higher accuracy.
2. Using Min-Max scaling to scale the training dataset and using the same Min and Max values from the training set scale the test dataset ($\frac{X-X_{min}}{X_{max}-X_{min}}$).
3. Using mini-batch gradient descent to optimize the loss function: "categorical cross-entropy" on the training dataset. Please run 50 epochs with batch size of 200. Also record the loss value for each of the 50 epochs. Record the time for each epoch. Generate a epoch-loss plot, and time-loss plot.
4. Evaluate the model on the test dataset:
   - print classification accuracy.
   - Print the 10-class confusion matrix.

3.5 Task 5: Variational Auto Encoder

1. Build a variational autoencoder with the following specifications (in this one you have a little more flexibility):
   - Should have at least two convolution layers in the encoder and 2 deconvolution layers in the decoder.
   - Latent dimension should be at least 5.
   - Loss should be either MSE or binary cross entropy.

2. Using Min-Max scaling to scale the training dataset and using the same Min and Max values from the training set scale the test dataset ($\frac{X - X_{min}}{X_{max} - X_{min}}$).
3. Using mini-batch gradient descent to optimize the loss function on the training dataset. Please run 50 epochs with batch size of 200. Also record the loss value for each of the 50 epochs. Record the time for each epoch. Generate a epoch-loss plot, and time-loss plot.
4. Qualitatively evaluate your model
   - Generate a set of clothes by randomly choosing 10 latent vectors and presenting the resulting images
   - Alter the network's architecture to produce some other results.

4 Additional Information

1. Exact learning parameters are not mentioned. You will need to select your own learning rate, momentum etc.
2. You should have a set of commend line variables which allow the user to run each of the tasks individually.

5 Report

You should submit a short PDF report with the following:

1. A short introduction to the problem.
2. Introduction to each network (including the ones you designed).
3. Results from each network. This should include the loss plots and the evaluation criteria
4. Conclusion
5. How to run your code.

6 Submission

You are required to submit one zip file with the code and your report.